Face localization by neural networks trained with Zernike moments and Eigenfaces feature vectors. A comparison

M. Saaidia[1,3] A. Chaari[1], S. Lelandais[1], V. Vigneron[1,2] M. Bedda[3]

[1]IBISC FRE CNRS 2873
Univ. Evry Val d'Essonne, FRANCE
{chaari, vvigne,saaidia}@iup.univ-evry.fr
s.lelandais@iut.univ-evry.fr

[2]Equipe SAMOS-MATISSE
Univ. Panthéon-Sorbonne, FRANCE
vigneron@iup.univ-paris1.fr

[3]Univ. de Annaba
Annaba, Algérie
mouldi_bedda@yahoo.fr

Abstract

Face localization using neural network is presented in this communication. Neural network was trained with two different kinds of feature parameters vectors; Zernike moments and Eigenfaces. In each case, coordinate vectors of pixels surrounding faces in images were used as target vectors on the supervised training procedure. Thus, trained neural network provides on its output layer a coordinate's vector (ρ, θ) representing pixels surrounding the face contained in treated image. This way to proceed gives accurate faces contours which are well adapted to their shapes. Performances obtained for the two kinds of training feature parameters were recorded using a quantitative measurement criterion according to experiments carried out on the *XM2VTS* database.

1. Introduction

In the last two decades, face detection received a growing attention by researchers concerned with Human-Machine communication. Thus, many face detection methods and algorithms were developed for images and videos which try to overcome different constraints like difference in brightness, pose and movement, faces appearance (glasses, beard and moustache), execution time, etc. These methods were useful for more complex techniques in Human-Machine communication like face recognition (identity check), gesture communication and face expression analysis and recognition.

According to Hjelmas and Low [1], face detection methods can be classified in 2 categories: "global approach" which consists in entirely seeking face and "components approach" which consists in finding the face through localization and regrouping of its components (eyes, nose...). They can be also classified according to face characteristics used like color, shape and movement.

Two *global* methods are presented and compared in this work. These two methods differ only on the first step which is the way to characterize the image to be treated. The first exploits geometrical characteristics of the face and the second uses projection on image sub-space variations. In the second step, a neural network (trained beforehand) uses the feature vector produced in the first step to output a coordinate's vector for pixels of the face's probable contour contained in the treated image. To make objective measure and comparison of methods performances we use a quantitative measurement criterion [2].

Geometrical moments, particularly Zernike ones, are used here for their capacity to compress the geometrical information, contained in the treated image, in a rather reduced parameters vector by projection of the image on an orthogonal basis [3]. In the same way, the Eigenfaces characterize the image by a reduced parameters vector representing variations of the treated image around an average image and according to some variation directions [4]. This compression characteristic makes them very adapted to the training of classifiers such as neural networks, which often need, on their input layer, feature vectors reduced in size but rather representative of the element subject to the classification. Zernike moments were particularly used for face recognition [5,6] and target recognition in general [7]. Eigenfaces were largely used in face detection and recognition directly [4,8] or through neural networks [9].

In section 2, we explain Zernike moments and Eigenfaces formulations. Section 3 develops the proposed way to their practical implementation. In section 4 we expose the measurement criterion and in section 5 experimental results are presented. Section 6 concludes.

2. Zernike moments and Eigenfaces Formulation

2.1. Zernike moments

Zernike moments are part of the geometrical moment's general theory. They were introduced initially by F. Zernike. Zernike moments are built on a set of orthogonal polynomials which allow construction of orthogonal base given by Eq. (1).

$$V_{n,m}(x,y)=V_{n,m}(\rho,\theta)=R_{n,m}(\rho).e^{j.m.\theta} \quad (1)$$

where:

$$\begin{cases} R_{n,m}(\rho)=\sum_{k=|m|}^{n}\frac{(-1)^{(n-k)/2}.(n+k)!}{(\frac{n-k}{2})!(\frac{k+m}{2})!(\frac{k-m}{2})!}.\rho^{k} \\ \rho=\sqrt{x^2+y^2} \quad \text{and} \quad \theta=arctg(y/x) \end{cases}$$

with $n \geq 0, m \neq 0, m < n, n-m < n$ and $n-k$ even.

$R_{n,m}(\rho)$ are the orthogonal radial polynomials, n is the order of the moment and m the repetition factor (the smoothness of the required details) at this order. ρ and θ are respectively the radius and the angle of function's treated point.

This base being orthogonal only inside the unit circle, the image to be projected must be mapped according to Eq.(2) which gives relations between the relative coordinates (i,j) of the initial image pixels and the new pixels coordinates (x_i, y_j) of the mapped one.

$$x_j = c+\frac{j.(d-c)}{(Q-1)} \quad \text{and} \quad y_i = d-\frac{i.(d-c)}{(P-1)} \quad (2)$$

where (P,Q) are dimensions of the image to be projected, i and j are indices of the point to be mapped and (c,d) defines couple of parameters allowing to map the function inside the unit circle (completely: $c=-1/\sqrt{2}$ and $d=-c$ or partially: $c=-1$ and $d=1$).

The projection of a numerical function $f(x_j, y_i)$ on the basis functions of Eq.(1) gives the Zernike moments $Z_{n,m}$ according to Eq.(3).

$$Z_{n,m}=\frac{n+1}{\pi}\sum_{x_j^2+y_i^2\leq 1}\sum f(x_j,y_i).V^{*}_{n,m}(x_j,y_i) \quad (3)$$

where: * denotes the *complex conjugate* of the function.

Traditional formulation of Zernike moments is very easy to implement but its computational time cost is very high. Researchers tried to overcome this major handicap by developing new formulations to enhance the speed computation [10,11]. The proposed algorithm in [11] (which is adopted here) has the advantage to preserve the same accuracy of computation as in the traditional formulation. To lead to this form of representation, previous equations are rewritten and reorganized in Eq.(4) which reduces the computation of Zernike moments for any image to the computation of a linear combination of $\beta_{n,m,k}$ and $X_{m,k}$.

$$\begin{aligned} Z_{n,m} &= \frac{n+1}{\pi}\sum_{x^2+y^2\leq 1}\sum\left(\sum_{k=|m|}^{n}\beta_{n,m,k}.\rho^{k}\right)e^{-j.m.\theta}.f(x_j,y_i) \\ &= \frac{n+1}{\pi}\sum_{k=|m|}^{n}\beta_{n,m,k}.\left(\sum_{x^2+y^2\leq 1}\sum e^{-j.m.\theta}.\rho^{k}.f(x_j,y_i)\right) \\ &= \frac{n+1}{\pi}\sum_{k=|m|}^{n}\beta_{n,m,k}.X_{m,k} \end{aligned} \quad (4)$$

where: $\beta_{n,m,k}=\frac{(-1)^{(n-k)/2}.(n+k)!}{(\frac{n-k}{2})!(\frac{k+m}{2})!(\frac{k-m}{2})!}$

2.2. Eigenfaces

"Eigenfaces" was the first method successfully used for face treatments like face detection and face recognition [4]. This method is based on the decomposition of the treated image according to some directions of variation around an average image. Decomposition is performed on a set of representative images of the characteristics to be classified. Based on Principal Components Analysis (PCA), Eigenfaces method uses SVD operation (Singular Values Decomposition) on a matrix containing a set of vectors, representing images, to determine their principal variety directions. In the case of face images, these main directions were called Eigenfaces.

To use Eigenfaces method, we first construct a projection space by operating SVD on the covariance matrix C_x given by Eq.(5). This operation gives the eigenvectors and eigenvalues of C_x arranged according to the variety directions importance.

$$C_x = E\left[(X-\mu_x).(X-\mu_x)^T\right], \quad (5)$$

where $X=(I_1,I_2,...,I_L)$ is an $K\times L$ face's matrix with L the number of faces and $K=P\times Q$ the dimension of the face vector I_i, ^{T} is the transpose operator, and the "average face" :

$$\mu_x(j)=\frac{1}{L}\sum_{i=1}^{K}I_i(j) \quad (6)$$

The set of eigenvectors obtained is used as projection space for images to be treated.

3. Methods implementation

Our contribution, with the goal to localize face in image, consists in three propositions:

1. The first one consists in the use of Zernike moments as training feature vectors for a neural network. Indeed, in addition to their capacity to compress geometrical

information of an image in a reduced vector, Zernike moments are not abstract parameters. Each one of them has a significance related to the statistical characteristics of the image which they represent such as the surface, the total mass center, mass centers in horizontal and vertical directions, horizontal and vertical symmetry, etc. Thus, the particular shape and contents of a face, geometrically rich by details of elements that it contains (eyes, mouth, etc), will be well represented in the set of Zernike moments.

2. The second one is the manner in which methods implementation will be done. We found that methods developed for face detection usually use rectangular or elliptical windowing research of the face on the treated image. This procedure gives non precise face contours and requires additional refinement operations. In our work we propose to train the neural network on target vectors which contain pixel coordinates obtained by manually delimiting faces in training images. This procedure will produce trained neural networks which provide precise and adapted face contours according to their shapes.

3. The last proposition consists of using a quantitative measurement criterion to record and compare the results obtained by each implemented method. The criterion is based on the compute of methods performances according to the number of pixels *correctly* and *wrongly* detected as belonging to the face in the treated image.

Figure 1 gives in a block diagram form the proposed face localization system.

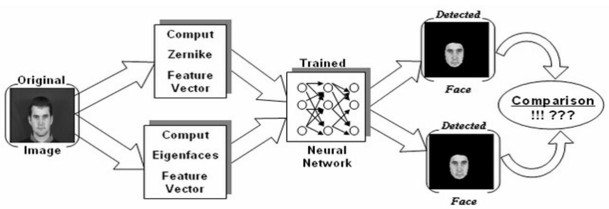

Figure 1: *General diagram of face localization system*

The implementation of our method is mainly based on the training phase which will be achieved in four stages:

(i) Computation of Zernike moments and Eigenfaces vectors for all the N images of the work database.

(ii) Construction of the training database by randomly taking N_1 images from the work database ($N_1 << N$) and their corresponding Zernike moments vectors Z_i and Eigenfaces vectors E_i.

(iii) Manual delimitation of the face area in each image of the training database by a set of points $\Omega_i = \{P_1, \cdots, P_d\}$ representing its contour.

(iv) Training of neural networks on the N_1 sets of couples (Z_i, Ω_i) or (Z_i, E_i).

Neural networks trained with Zernike feature vectors learn to extract statistical information contained in Zernike moments and in their interactions which are closely related to the area of the required face. Those trained with Eigenfaces feature vectors learn to identify the main variety directions introduced by face in training images.

To test and measure performances of the network obtained after training operation, we proceed, according to Figure 1, on all $N - N_1$ images remaining in the work database. Face localization procedure will be the same for the two methods compared in this work and will be done in two steps:

- During first step, an image is presented to a program which extracts Zernike or Eigenfaces feature vector.
- At the second step, a back-propagation neural network, beforehand trained, receives on its input layer the feature vector which was computed in first step. In response, it gives on its output layer a coordinate's vector for a set of points representing the probable face contour contained in the treated image.

Results obtained by each method according to equivalent parameters and for the same images are then quantitatively compared.

4. Quantitative measurement criterion

To give an objective appreciation of results given by the studied methods, we propose a new way to calculate the detection rate based on the relation between the number of pixels correctly and wrongly detected as pixels of the face and the number of face pixels in each treated image. To do so, all testing database images were manually segmented in three regions like it is shown in Figure 2.

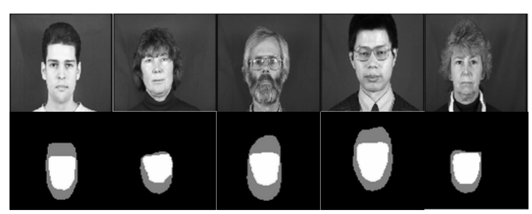

Figure 2: *Examples of regions definition. Top: original image, Bottom: mask of regions*

First region (white one on the masks of Figure 2) contains the W pixels which represent the essential components of the face (brows, eyes, nose, mouth and surrounding pixels). The second region (grey one) contains pixels surrounding the first region and belonging to the face. The last region contains all the B pixels of the image which do not belong to the face. For the detection system, the first region is one which has to be contained imperatively in the resulting contour and the third one has to be imperatively discarded from it. The second region is optional and has no effect on the computed results. We define two types of measures; Good detection rate (Gdr) and Quality detection rate (Qdr).

$$Gdr = \frac{W_1}{W}.100 \quad \text{and} \quad Qdr = \left(\frac{W_1}{W} - \frac{B_1}{A-B}\right)100 \qquad (7)$$

where W_1 and B_1 are respectively the number of pixels *correctly* and *wrongly* detected as belonging to the face and A is the number of all pixels of the image.

The Gdr measures how many pixels from the essential parts of the face are detected. The Qdr gives a more strict measure of face detection taking pixels of image that wrongly detected as belonging to the face into account. These two rate measures are complementary. Having only Gdr we don't know how many pixels are wrongly detected as belonging to processed face. In the same way, having Qdr only we don't know how many pixels belonging to the face are not detected. On Figure 3 we illustrate this fact on some examples with recorded Gdr and Qdr values.

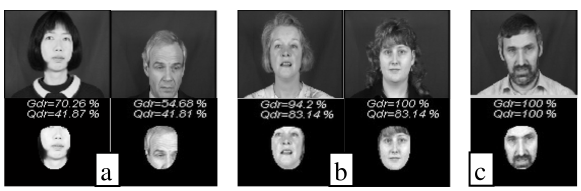

Figure 3: *Top: original images. Bottom:* complementary *relation between Gdr and Qdr measures*

On Figure 3.a we have the same bad Qdr (about 41%), with two different Gdr(54% and 70%). On Figure 3.b it's the same situation for a good Qdr (about 83%) with two different Gdr values (94% and 100%). To finish, we give on Figure 3.c an example of a face perfectly detected with Gdr and Qdr at 100%. Thus, to have a correct appreciation of recorded results, each one of Gdr and Qdr has to be computed. Best results are obtained when they are both closest to 100% with minimum difference between them.

5. Experimental results

In order to check the validity of our proposed method and to compare methods performances studied here, experimental studies were carried out on the *XM2VTS* images database [12]. This extended database contains 4 recordings of 295 subjects taken over a period of 4 months with rotating head shot in vertical and horizontal directions. Images are colored and in ppm format. In our experiments we brought some transformations to original images like change to GIF format (more compressed) and the use of luminance information only (grey scale images) to compute the Zernike moments and Eigenfaces vectors.

To obtain a training database we take randomly 15 subjects with their first 3 different recordings, so that gives 45 examples of couples (Z_i, Ω_i) and (Z_i, E_i) to train neural networks. To have a precise and rather general idea on method performances, we carried out the construction of 20 training databases always by randomly taking examples from the first 3 recordings of the database. For each test, we compute the average values of Gdr and Qdr and their Standard deviations (Std) σ. Neural networks trained and used in our experiments have 60 neurons on their output layers so they provide 30 coordinate pairs (ρ, θ) for 30 pixels. This number was experimentally chosen to be sufficient to surround face region in the treated image.

Our experiments aimed at the study of behavior of the two methods according to the training database, training vectors dimension and the neural network complexity.

5.1. General results

First, we present in Figure 4 an example of results given by two different trained neural networks applied to the 295 images of the fourth database recording. The first neural network was trained with Zernike moments feature vectors. The second was trained with Eigenfaces.

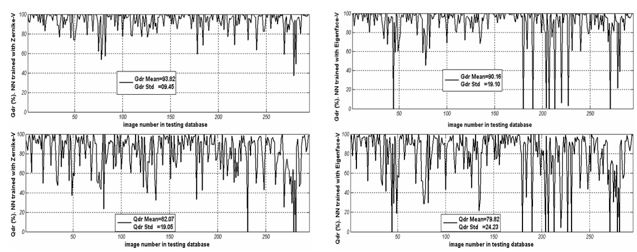

Figure 4: *Gdr and Qdr curves for the images of the fourth Database recording. Neural network trained with:(left) Zernike moments, (right) Eigenfaces*

These rates are the best ones, according to different training databases with feature vector dimension equal to 22. The resulting rates show that in the case of Zernike moments only few faces were incorrectly detected. Most of the images were correctly treated indicating good generalization performances. However, less performance results were recorded in the case of Eigenfaces.

Table 1: *Figure's 4 results summary (Ni: images Number)*

	Zernike Training		Eigenfaces Training	
	Ni	(Ni/295) %	Ni	(Ni/295) %
Gdr<70%	**11**	**3.72**	**23**	**7.79**
Qdr<70%	**60**	**20.33**	**74**	**25.08**
Gdr *Mean / Std*	**93.82 / 9.45**		**90.16 / 19.10**	
Qdr *Mean / Std*	**82.07 / 19.05**		**79.82 / 24.23**	

Table 1 shows that in the first case (Zernike moments training) 80% of images have Gdr and Qdr greater than 70%, however only 75% in the case of Eigenfaces. This performances superiority is also apparent by comparing the Qdr averages and Std computed for all images.

On Figure 5 we give some examples of good detected faces from the testing database. We chose images with some faces variability in terms of *position*, *color*, *pose*, *size* and *gender*. Results illustrate the difference between Gdr and Qdr measures and also between performances of the two compared methods.

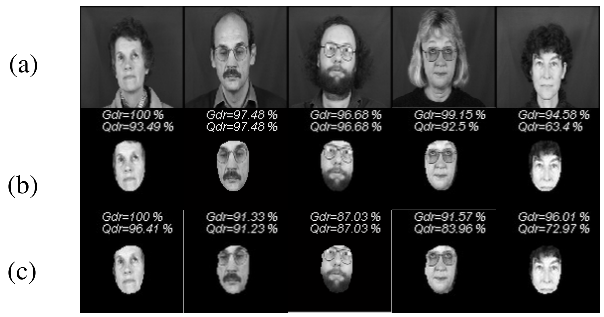

Figure 5: *Examples of face detection.(a): original images (b): faces detected with Zernike training vectors (c): faces detected with Eigenfaces training vectors*

5.2. Training database influence

To study training database influence on each of the presented methods and also to obtain more reliable performances comparing between them, experiments were carried out on the 20 training databases randomly constructed.

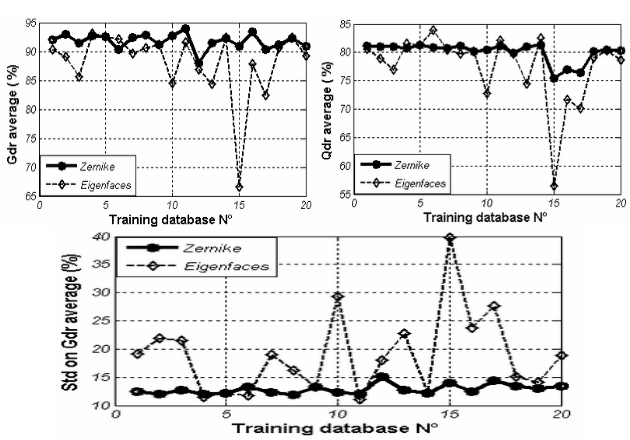

Figure 6: Gdr, *Qdr and Std variations related to the Training feature vectors (Zernike or Eigenfaces) and databases (1,2, …, 20). Top: Gdr and Qdr averages, Bottom: Std*

Results given on Figure 6 were obtained by training, testing and measuring performances of a neural network for each one of the twenty training databases with the same fixed parameters. Input, hidden and output layers have respectively 6, 10 and 60 neurons with a *sigmoid* activation function for the hidden layer and *linear* activation function for the output layer. "Resilient propagation" was used as neural training function.

According to Gdr, Qdr and Std reported by curves in Figure 6, we can say that training with Zernike feature vectors gives best results than training using Eigenfaces ones. Indeed, for the first case Gdr averages are greater than 90% (up to 94%) for almost the totality of the training databases and Qdr averages are about 80%. In the same way, low Std values show good generalization performances on images of the testing database. Results obtained for Eigenfaces training show a greater sensitivity to the training databases and a bad generalization performance (big values of Std) for most of them. Indeed, up to 28% of difference in Qdr and up to 40% in Std are recorded for neural networks trained with Eigenfaces feature vectors, according to the training database. For those trained with Zernike moments, this difference is only about 6% and Std values are no more than 15%.

5.3. Feature vectors size influence

Feature vectors size has significance related to the quantity of image information included and compressed by these vectors. Zernike vectors size is controlled by parameters m and n while that of Eigenfaces is controlled by the size of the projection space constructed. Moreover, vectors size determines the number of neurons in the neural network input layer and hence, its complexity.

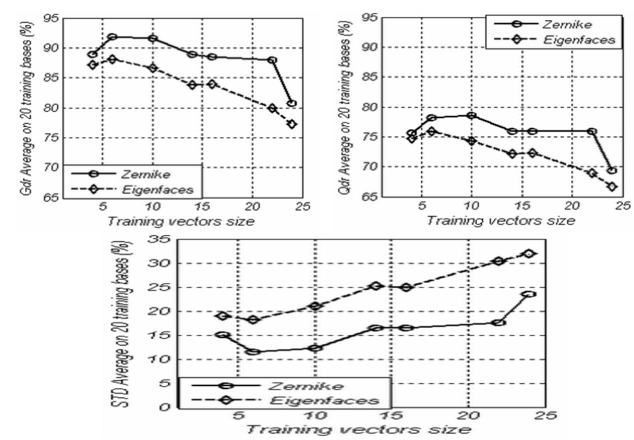

Figure 7*: Gdr and Qdr averages and Std variations computed on 20 independent training databases with 7 vectors size: 4, 6, 10, 14, 16, 22 and 24*

On Figure 7 we give the variation curves of Gdr and Qdr averages and the Std, computed on the totality of the results given by the twenty neural networks trained on the twenty training databases, according to 7 different size values of the feature vectors.

Here also, we can see that Zernike moments vectors provide best average results for the 7 cases studied.

For the two kinds of training feature vectors, best results were obtained for vector sizes between 6 and 10. For other vector sizes the quality of localization decreases considerably. In the case of sizes lower than 6, this will be due to insufficient information brought by vector parameters. For those larger than 10, neural networks become more complex and thus convergence is more difficult.

6. Conclusion

Face localization using neural networks and a new way to train them were presented in this communication. We compared results given by neural networks trained with Zernike moments feature vectors and those trained with Eigenfaces ones, according to a proposed quantitative measurement criterion which allows an automatic measure and appreciation of results.

Recorded results of quality detection and capacity of generalization demonstrate the superiority performances given by the neural networks trained with Zernike moments feature vectors. Good localization rates, up to 94%, were achieved and accurate contours adapted to the face shapes were obtained.

These results demonstrate also the high sensitivity of neural networks trained with Eigenfaces to the training database. A difference about 28% was recorded for them while only 6% of difference for those trained with Zernike moments.

For the two methods, best results were obtained for the vectors size 6. Decreasing evolution of Gdr and Qdr averages is observed for sizes lower than 6 and greater than 10 where, in the same time, the Std values increase considerably. This indicates a decrease in the generalization capacity of the trained neural networks.

Method using Zernike moments feature vectors can be extended to face components detection and object detection in general. Method implementation performances can be improved by judicious choices on the training database size and contents and also by adapted parameters of training vectors and neural network.

Acknowledgment

M. SAAIDIA is supported by a national grant from the ministry MESRS of the Algerian state.

References :

[1]. E. Hjelmas et B. K. Low. “Face detection : A survey” Computer Vision and Image Understanding, vol. 83, no. 3, pp. 236-274, 2001.

[2]. M. Saaidia, S. Lelandais, V. Vigneron and M. Bedda, "Face detection using Zernike moments. Evaluation by quantitative measurement creterion", WSEAS Transactions on Signal Processing., Issue 2, Vol. 3, pp 226-232, 2007.

[3]. M.R. Teague, “image analysis via the general theory of moments”, J. Opt. Soc. Amer., vol. 70, pp. 920-930, 1980.

[4]. M. Turk and A. Pentland, “Eigenfaces for recognition”, Journal of cognitive Neuroscience, vol. 3, pp. 71-86, 1991.

[5]. J. Haddadnia, M. Ahmadi, and K. Faez, “An Efficient Feature Extraction Method with Peudo Zernike Moment in RBF Neural Network Based Human Face Recognition System”, EURASIP JASP, vol. 9, pp. 890-891, 2003.

[6]. S. Alirezaee, M. Ahmadi, H. Aghaeinia, and K. Faez, “A weighted pseudo-zernike feature extractor for face recognition”, IEEE, ICSMC, vol. 3, pp. 2128-2132, 2005.

[7]. C. Rosenberger, A. Rakotomamonjy, B. Emile, "Generic target recognition", EUSIPCO vienne, pp. 1613-1616, 2004.

[8]. Y.L.Y.Zhuejie, “Face recognition with Eigenfaces”, Proc. IEEE Intl. Conf. Industrial Technologies, pp. 434-438, 1994.

[9]. M. Rizon, M.H. Firdus, P. Saad, S. Yaacob, M.R. Mamat, A.Y.M. Shakkaf, A.R. Saad, Hdesa and M. Karthigayan, “ Face recognition using Eigenfaces and Neural Networks’’, American Journal of Applied Sciences, vol. 2, pp. 1872-1875, 2006.

[10]. J. Gu, H. Z. Shua, C. Toumoulinb, and L. M. Luoa, “A novel algorithm for fast computation of Zernike moments”, Pattern Recognition, vol 35, pp. 2905-2911, 2002.

[11]. G. Amayeh, A. Erol, G. Bebis, and M. Nicolescu, “Accurate and efficient computation of high order zernike moments”, First ISVC, Lake Tahoe, NV, USA, pp. 462–469, 2005.

[12]. K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, ’’XM2VTSDB: The Extended M2VTS Database’’ Proc. AVBPA’99 Washington DC, 1999.